

JP9305436

Biblio Page 1

esp@cenet

PROGRAM DEVELOPING SYSTEM

Patent Number: **JP9305436**

Publication date: **1997-11-28**

Inventor(s): **KURAHASHI KEIICHI**

Applicant(s): **NEC CORP**

Requested Patent: **JP9305436**

Application **JP19960148361 19960517**

Priority Number(s):

IPC Classification: **G06F11/28 ; G06F9/45**

EC Classification:

Equivalents: **JP2812302B2**

Abstract

PROBLEM TO BE SOLVED: To shorten a program development time.

SOLUTION: A source program is inputted, line by line, and analyzed (steps 201 and 202).

When one line of the analyzed source program is the end of the source program, the process is finished (steps 203 and 208). One line of the analyzed source program is a debug command, debug command information is generated and outputted (steps 204, 207, and 208). When one line of the analyzed source program is an object of object output as a line for outputting a machine word or segment line, an object, etc., is generated and outputted.

Data supplied from the **esp@cenet** database - 12

(19)日本国特許庁 (JP)

(12) 特許公報 (B2)

(11)特許番号

第2812302号

(45)発行日 平成10年(1998)10月22日

(24)登録日 平成10年(1998)8月7日

(51)Int.Cl.
G06F 11/28
9/45

識別記号
315

F1
G06F 11/28
9/44

815A
822E

請求項の数 2(全 9 頁)

(21)出願番号 特願平8-148981
(22)出願日 平成8年(1996)5月17日
(65)公開番号 特開平9-305436
(43)公開日 平成9年(1997)11月28日
(54)審査請求日 平成8年(1996)5月17日

(73)特許権者 000004237
日本電気株式会社
東京都港区芝五丁目7番1号
(72)発明者 倉橋 啓一
東京都港区芝五丁目7番1号 日本電気
株式会社内
(74)代理人 弁理士 五十嵐 駿三

審査官 深沢 正志

(56)参考文献 特開 平4-387946 (JP, A)
特開 平7-319730 (JP, A)

(58)調査した分野(Int.Cl., DB名)

G08F 11/28
G08F 9/45

(54)【発明の名称】 プログラム開発装置

1

(57)【特許請求の範囲】

【請求項1】 ソースプログラムファイル(4)よりソースプログラムを入力する手段(11)と、
該入力されたソースプログラムを解析する解析手段(12)と、
該解析した結果からソースプログラムにオブジェクト出力対象があるときにはオブジェクト情報を生成して出力する手段(13)と、
前記解析した結果からソースプログラムにデバックコマンドがあるときにはデバッグコマンド情報を生成して出力する手段(14)とを具備するプログラム開発装置。

【請求項2】 さらに、

前記生成されたオブジェクト情報の結合、配置を行い、
該オブジェクト情報の補正を行って出力し、前記デバッグコマンド情報を補正を行って出力するリンク部(2)

2

と、

前記ソースプログラムと前記リンク部から補正されたオブジェクト情報との対応付けを行い、前記リンク部から補正されたデバッグコマンド情報によりデバッグコマンドの設定を行い、該デバッグコマンドを前記オブジェクト情報に対して実行するデバッカ部(3)とを具備する請求項1に記載のプログラム開発装置。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 本発明は言語処理プログラムプログラム開発装置に関する。

【0002】

【従来の技術】 従来のプログラム開発装置を図7を参照して説明する。プログラム開発装置は、ソースプログラムをオブジェクト情報(オブジェクトプログラムまたは

オブジェクトモジュール)に変換するアセンブラー部1、複数のオブジェクト情報を結合してロードモジュールにするリンク部2、及びロードモジュールのデバッグを行うデバッガ部3よりなる。以下、各部について説明する。

【0003】アセンブラー部1においては、ソースプログラムファイル4からソースプログラムが入力部11に入力され、機文解析部12によりこのソースプログラムを解析する。この解析の結果、オブジェクト出力部13はオブジェクト情報及び行番号情報をオブジェクトファイル5に出力する。

【0004】リンク部2においては、オブジェクトファイル5からオブジェクト情報、行番号情報が入力部21に入力され、結合配置部22によりこのオブジェクト情報、行番号情報の結合、配置を行い、絶対アドレスを確定する。この絶対アドレスの確定の結果、ロードモジュール出力部23はロードモジュール及び行番号情報をロードモジュールファイル6に出力する。

【0005】デバッガ部3においては、ロードモジュールファイル6からロードモジュール行番号情報が入力部31に入力され、デバッグコマンド7を設定する。デバッグ部32は、ロードモジュール、行番号情報により、ソースプログラムとオブジェクト情報との対応付けを行い、デバッグコマンド7によりデバッグを行う。たとえば、ソースプログラムの100行目でブレークをする場合、デバッグコマンドにより100行目でブレークを行うことを指示し、この結果、デバッグ部32において、プログラムの実行が開始し、上記デバッグコマンドにより対応するオブジェクト情報をプログラムの実行を停止する。このように、ソースプログラムと直接関係する行番号情報をオブジェクトファイル5、ロードモジュールファイル6に出力し、デバッガ3においてこの行番号情報を利用している。すなわち、デバッガ3において、ある行でブレークポイントを設定する等を、デバッグの度にデバッグコマンドで設定し、このとき、行番号情報と絶対アドレスとの対応付けにより設定する。

【0006】

【発明が解決しようとする課題】しかしながら、図7の従来のプログラム開発装置においては、ソースプログラムの変更毎にデバッグコマンドの再設定を行わなければならず、この結果、デバッグに相当の時間を要し、プログラムの開発時間が多大になるという課題がある。なお、ソースプログラムレベルでデバッグを行うものがあるが(参照:特開平3-53347号公報、特開平4-169941号公報)、ステップ動作を行わない場合、再コンパイルが必要となる欠点がある。従って、本発明の目的は、プログラム開発時間を短縮できるプログラム開発装置を提供することにある。

【0007】

【課題を解決するための手段】上述の課題を解決するた

めに本発明は、入力されたソースプログラムを解析した結果、オブジェクト出力対象があるときには、オブジェクト情報を生成して出力し、デバッグコマンドがあるときには、デバッグコマンド情報を生成して出力する。すなわち、ソースプログラムで指定されたデバッグコマンドをデバッグコマンド情報として生成することにより、デバッグ時のコマンドの再設定は不要となる。また、デバッグコマンドの設定はプログラムのオブジェクトとは別に生成されるので、デバッグ時のプログラムと最終プログラムの同一性が確保される。

【0008】

【発明の実施の形態】図1は本発明に係るプログラム開発装置の実施の形態を示すブロック図である。図1においては、図7のプログラム開発装置に対して、デバッグコマンド7の代りに、デバッグコマンド情報ファイル8、9を附加してある。また、アセンブラー部1には、デバッグコマンド情報をデバッグコマンド情報ファイル8に出力するデバッグコマンド情報出力部14を追加してある。また、リンク部2には、デバッグコマンド情報を補正してデバッグコマンド情報ファイル8に出力するデバッグコマンド情報出力部24を附加してある。なお、ロードモジュール出力部23はオブジェクト情報、行番号情報の補正を行ってロードモジュールファイル6に出力する。さらに、デバッガ部3の入力部31はデバッグコマンド情報ファイル9からデバッグコマンドが入力される。以下、図1の各部の動作を説明する。

【0009】図2は図1のアセンブラー部1の動作を示すフローチャートである。始めに、ステップ201では、ソースプログラムファイル4からソースプログラムの1行を入力する。次に、ステップ202では、入力されたソースプログラムの1行を解析する。次いで、ステップ203では、解析されたソースプログラムの1行がソースプログラムの終了か否かを判別し、ステップ204では、解析されたソースプログラムの1行がデバッグコマンドか否かを判別し、ステップ205では、解析されたソースプログラムの1行がオブジェクト出力対象か否かを判別する。

【0010】従って、解析されたソースプログラムの1行がソースプログラムの終了であれば、ステップ203からステップ206に進み、処理を終了する。解析されたソースプログラムの1行がデバッグコマンドであれば、ステップ204からステップ207に進み、デバッグコマンド情報を生成を行う。すなわち、図6に(A)示すように、コマンド種別、行番号、相対アドレス及びセグメント番号を生成し、ステップ208にてデバッグコマンド情報をデバッグコマンド情報ファイル8に出力する。そして、ステップ201に戻り、ソースプログラムの次の行を入力する。解析されたソースプログラムの1行が機械語を出力する行あるいはセグメント行であるオブジェクト出力対象であれば、ステップ205からス

(3)

特許2812302

6

5
テップ209、210に進む。すなわち、ステップ209では、オブジェクトあるいはセグメント情報、行番号情報を生成し、ステップ210にてオブジェクトファイル5に出力する。そして、ステップ201に戻り、ソースプログラムの次の行を入力する。また、ステップ205にてオブジェクト出力対象でなければ、やはり、ステップ201に戻り、ソースプログラムの次の行を入力する。

【0011】図5のソースプログラムを例として図2の動作を説明する。モジュールAの1行目の"C1CSE" 10 G"をステップ202にて解釈すると、オブジェクト出力対象のセグメント行であるので、ステップ205からステップ209、210に進む。すなわち、セグメント情報、行番号情報を生成してオブジェクトファイル5に出力する。モジュールAの2行目の"NOP"をステップ202にて解釈すると、オブジェクト出力対象の機械語を出力する行であるので、ステップ205からステップ209、210に進む。すなわち、オブジェクト、行番号情報を生成してオブジェクトファイル5に出力する。

【0012】10行目の"SETBR"をステップ202にて解釈すると、デバッグコマンドであるので、ステップ204からステップ207、208に進む。すなわち、図6の(B)に示すように、デバッグコマンド情報として、コマンド種別を"ブレークコマンド"、行番号を"10"、相対アドレスを"12H"、セグメント番号を"1"として生成し、デバッグコマンド情報ファイル8に出力する。50行目の"END"をステップ202にて解釈すると、ソースプログラムの終了であるので、ステップ203からステップ208に進み、処理を終了する。

【0013】図3は図1のリンク部2の動作を示すフローチャートである。ステップ301では、オブジェクトファイル5及びデバッグコマンド情報ファイル8からオブジェクト、デバッグコマンド情報を入力する。ステップ302では、入力されたオブジェクト情報の結合、配置を行う。ステップ303では、オブジェクトかデバッグコマンド情報かの判別を行う。この結果、オブジェクトであれば、ステップ304、305に進み、デバッグコマンド情報であればステップ306、307に進む。ステップ304では、オブジェクト情報、行番号情報の補正を行い、ステップ305にてロードモジュールファイル8に出力する。ステップ306では、デバッグコマンド情報の補正を行い、ステップ307にてデバッグコマンド情報ファイル9に出力する。ステップ308では、すべての処理が終了したか否かを判別し、すべての処理が終了していればステップ309に進み、処理を終了する。

【0014】たとえば、図5のモジュールAのセグメント"C1"が100番地に配置された場合を考える。C

の場合、デバッグコマンド情報であるので、ステップ303からステップ308に進む。この結果、セグメント番号"1"によりセグメント"C1"を限定し、確定したベースアドレス"100H"を得、相対アドレス"12H"に加算し、図6の(C)に示すとく、絶対アドレス"112H"を得る。

【0015】図4は図1のデバッグ部3の動作を示すフローチャートである。ステップ401では、ロードモジュールファイル6からオブジェクト情報と、行番号情報を入力し、ステップ402にて、ソースプログラムとオブジェクトとの対応付けを行う。次に、ステップ403では、デバッグコマンド情報ファイル9からデバッグコマンド情報を入力し、ステップ404にてデバッグコマンドの設定を行う。次に、ステップ405では、デバッグコマンドの行に対応したオブジェクトの位置でコマンドの実行を行う。次に、ステップ408にてデバック終了か否かを判別し、この結果、デバック終了であればステップ407に進み、処理を終了する。他の場合、ステップ405に進む。

【0016】たとえば、図6の(C)に示す場合、コマンド種別"ブレークコマンド"が示すモジュールAの行番号10行目に対応した絶対アドレス"112H"の位置でプログラムの実行を停止する。

【0017】このように、ソースプログラムにおいてデバックコマンドを指定することによりデバッグコマンド情報を生成してデバッグを行うことにより、ソースプログラムを変更した場合でも、デバッグコマンドの再設定は不要ない。また、デバックコマンドの設定はプログラムのオブジェクトとは別に生成されるので、デバッグ時のプログラムと最終プログラムとの同一性が確保される。

【0018】

【発明の効果】以上説明したように本発明によれば、デバッグ時のコマンドの再設定は不要となり、また、デバッグ時のプログラムと最終プログラムとの同一性が確保されるので、プログラム開発時間を大幅に短縮できる。

【図面の簡単な説明】

【図1】本発明に係るプログラム開発装置の実施の形態を示すブロック図である。

【図2】図1のアセンブラー部の動作を示すフローチャートである。

【図3】図1のリンク部の動作を示すフローチャートである。

【図4】図1のデバッグ部の動作を示すフローチャートである。

【図5】プログラムの一例を示す図である。

【図6】デバッグコマンド情報を示す図である。

【図7】従来のプログラム開発装置を示すブロック図である。

50 (符号の説明)

(4)

特許2812302

8

1—アセンブラ部
2—リンク部
3—デバック部
4—ソースプログラムファイル

* 5—オブジェクトファイル
6—ロードモジュールファイル
7—デバックコマンド
* 8, 9—デバックコマンド情報

【図5】

ソースプログラムの一例

```
エディタヘルプ
```

```

1: C1 CSE0
2: 0000 NOP
~  

10: 0018 853752
~  

30: 870

```

【図6】

(A) デバッグコマンド情報

コマンド種別	ブレークコマンド
行番号	10
絶対アドレス	12H
セグメント番号	1

(B) コマンド種別 ブレークコマンド

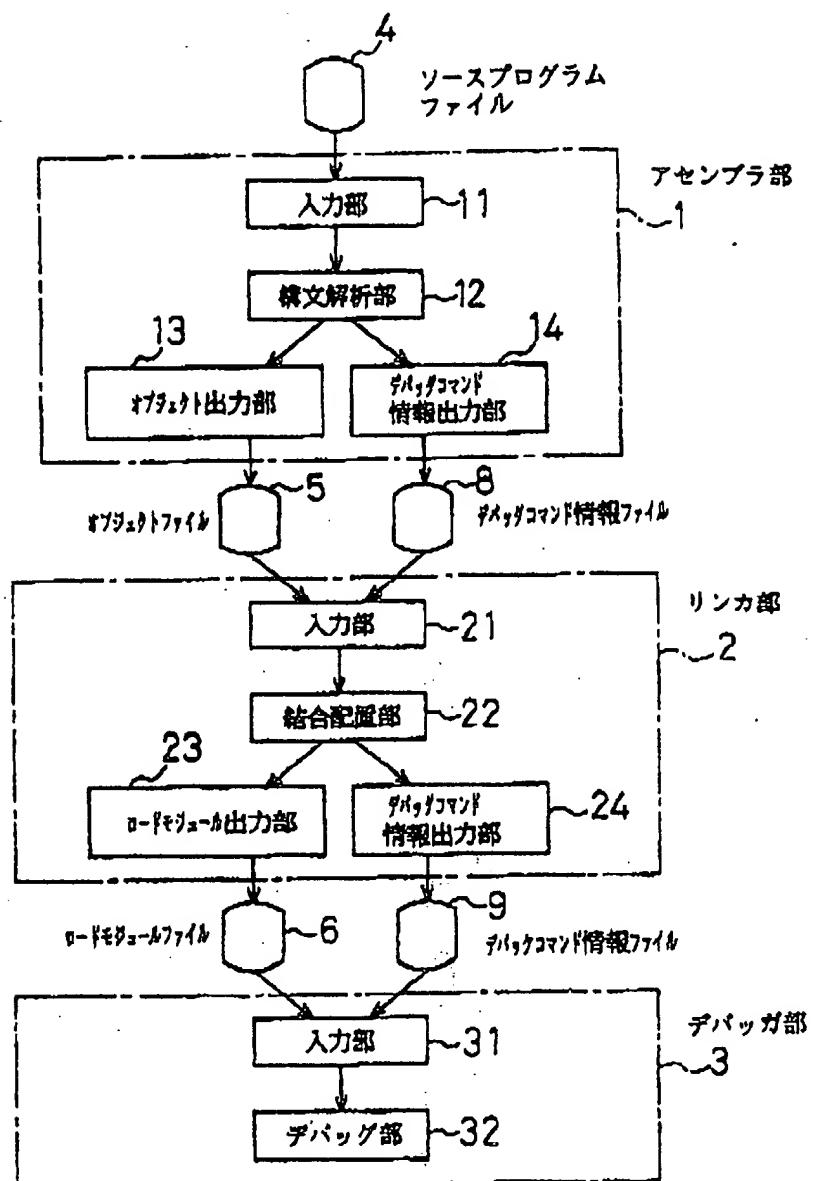
行番号	10
絶対アドレス	12H
セグメント番号	1

(C) コマンド種別 ブレークコマンド

行番号	10
絶対アドレス	118H
セグメント番号	1

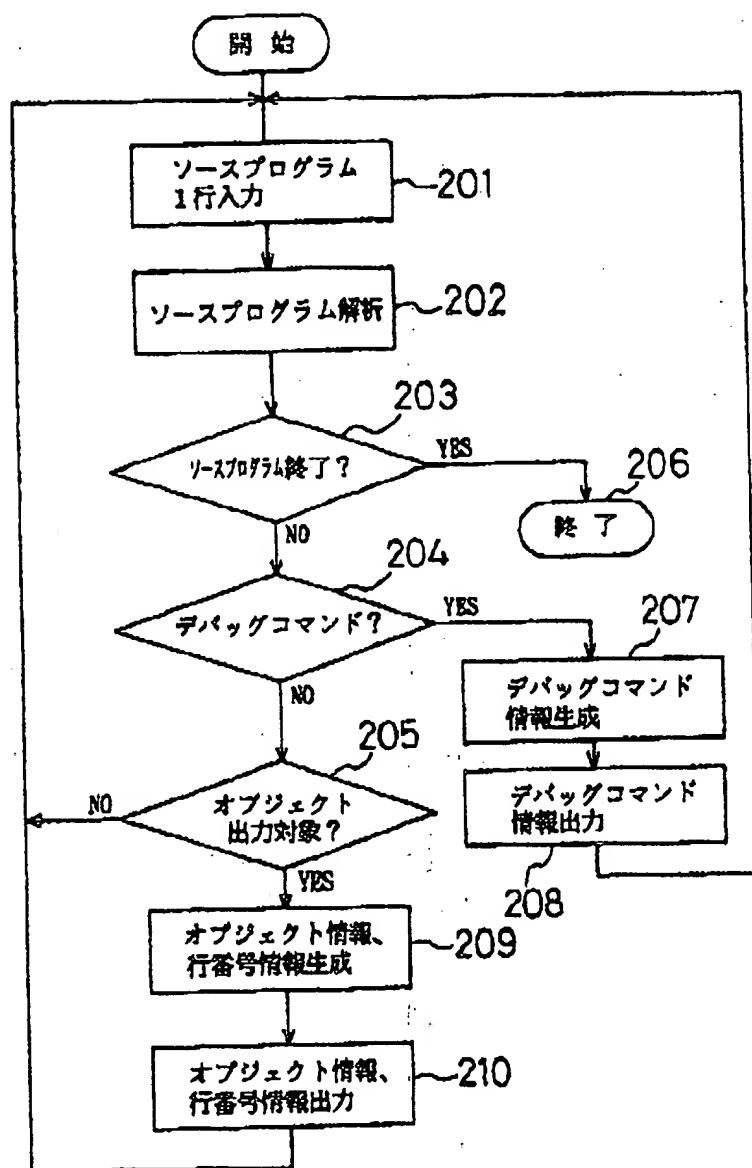
【図1】

本発明に係るプログラム開発装置



[図2]

図1のアセンブラー部の動作

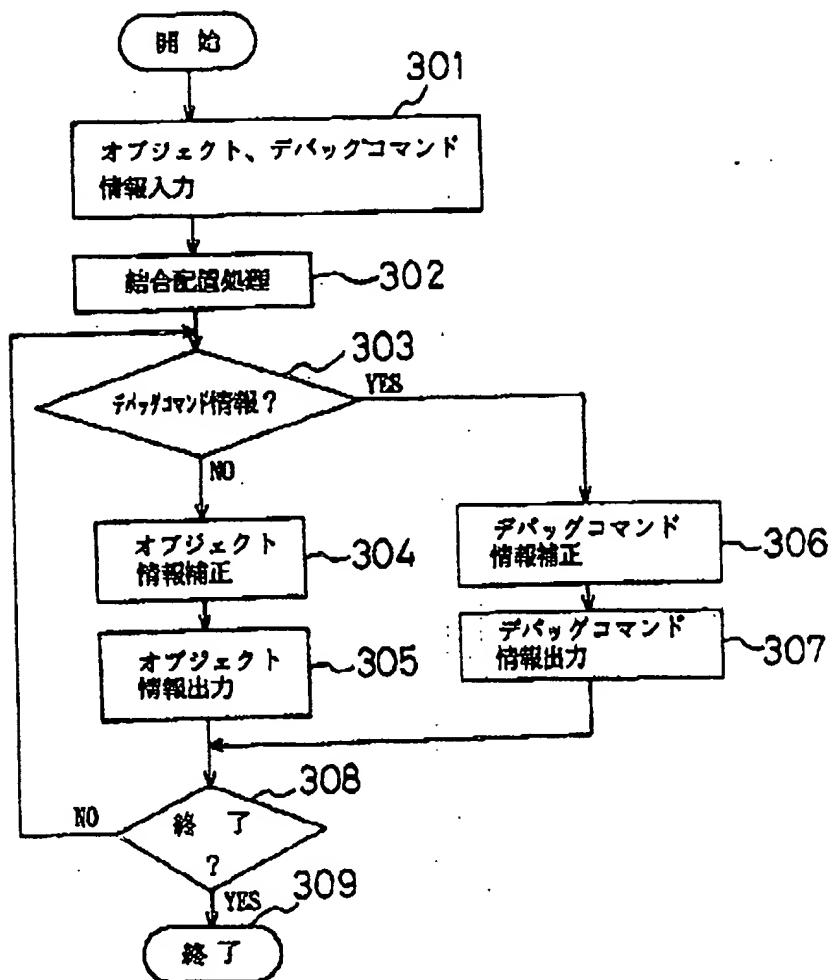


(7)

特許2812302

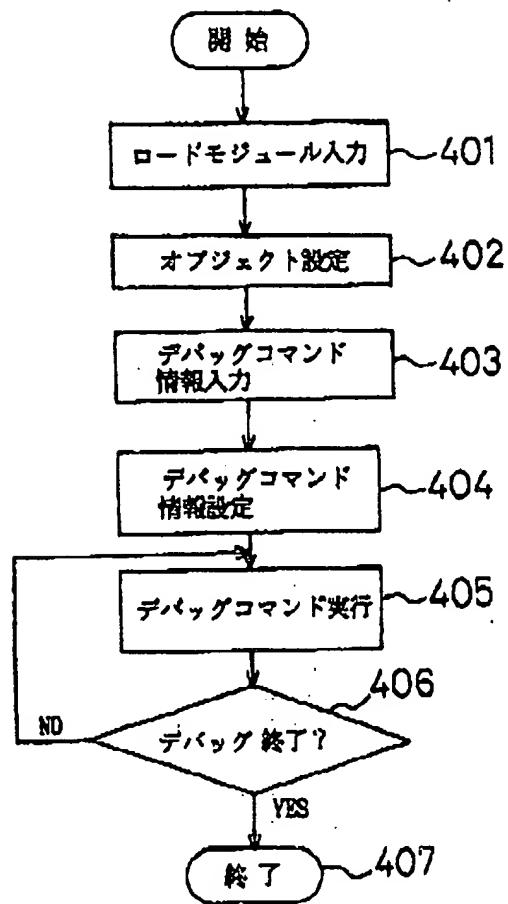
〔図3〕

図1のリンク部の動作



〔図4〕

図1のデバッグ部の動作



【図7】

従来のプログラム開発装置

